PREPARED FOR SUBMISSION TO JINST (accepted for publication)

New perspectives on segmented crystal calorimeters for future colliders

https://arxiv.org/abs/2008.00338

M.T. Lucchini,[a,1] **W. Chung,**[a] **S.C. Eno,**[b] **Y. Lai,**[b] **L. Lucchini,**[c] **M. Nguyen,**[a] **C.G. Tully**[a]

[a] *Princeton University, Princeton, New Jersey, USA*
[b] *University of Maryland, College Park, Maryland, USA*
[c] *Fondazione Bruno Kessler, Trento, Italy*

E-mail: marco.lucchini@cern.ch

Physics Performance of the New Concept with Crystal ECAL and Dual Readout HCAL

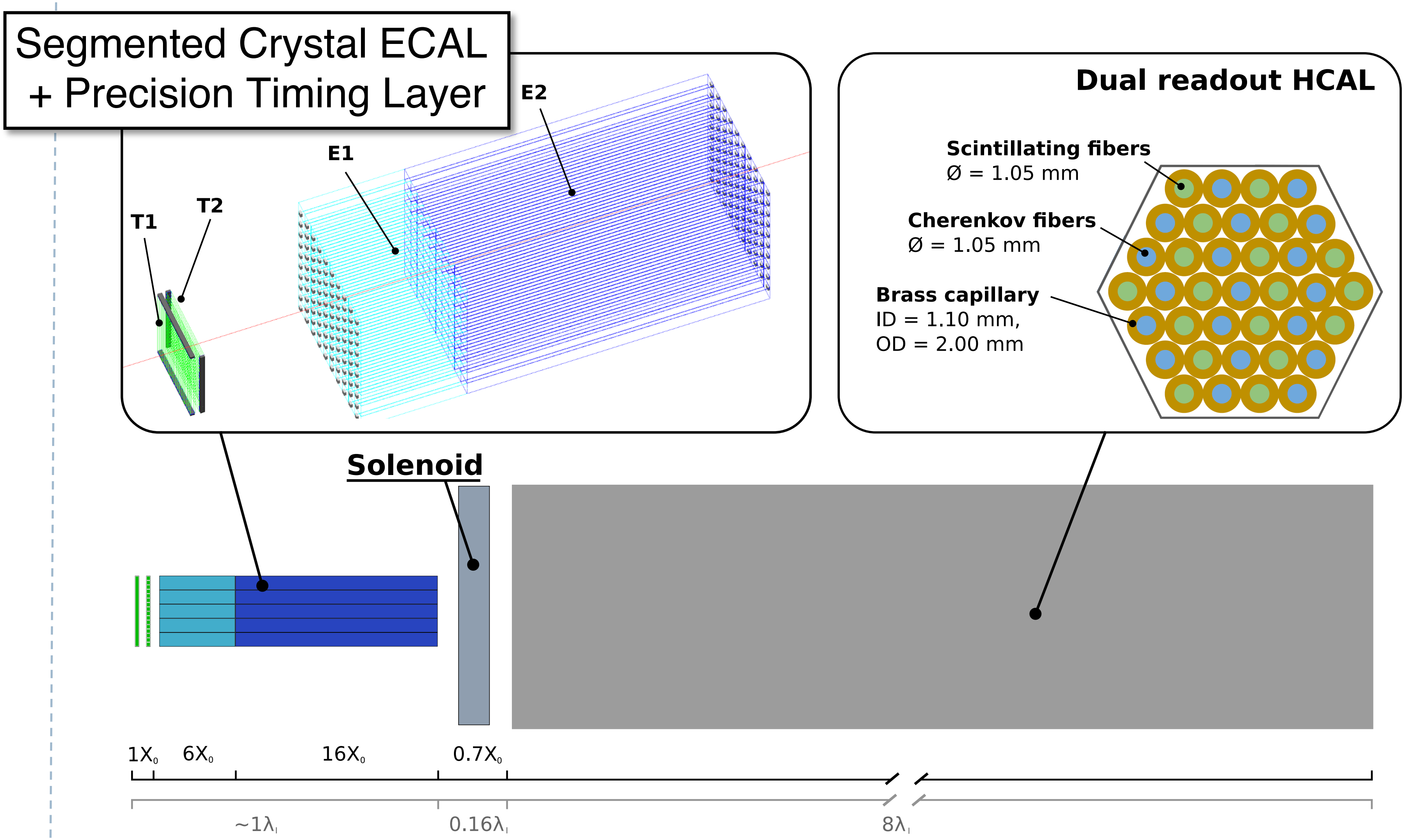
Chris Tully (Princeton)

The 2020 International Workshop on High Energy Circular Electron Positron Collider

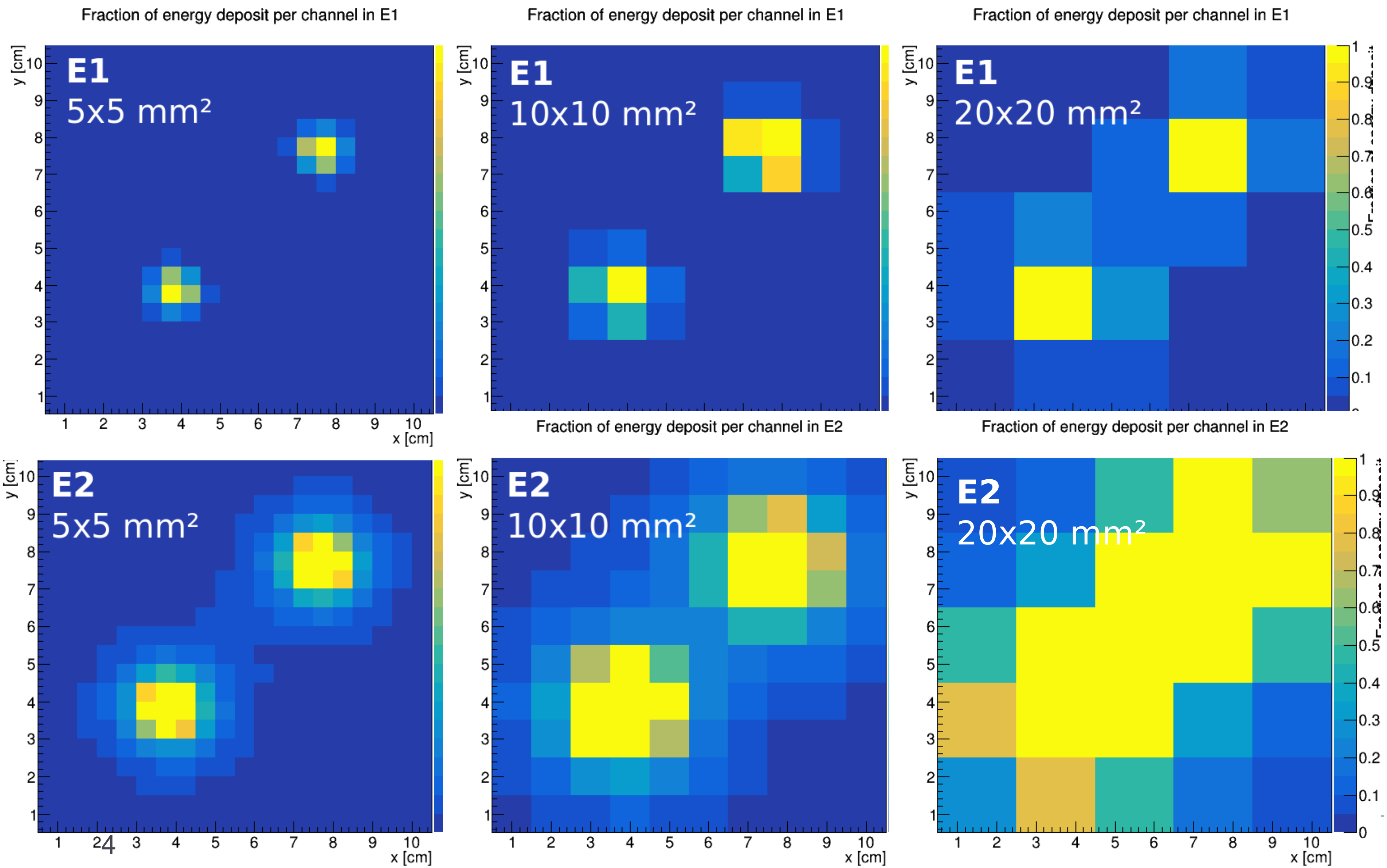
What is in the paper?

- **A calorimeter design that has the highest energy resolution for both photons and neutral hadrons**
  - Implemented a working solution for the problem of crystals destroying hadronic calorimeter resolution
  - Solved some practical issues related to the solenoid and cost-performance optimization
- **Demonstrated the range of tracker parameters and high EM resolution that bring Z→ee up to ~80% the Z→$\mu\mu$ recoil resolution**
- **Introduced π^0 photon pre-clustering performance benchmarks for EM resolution**
- **Calorimeters have the highest interaction cross-sections and therefore high intrinsic potential for particle ID**
  - We explore e/pi (~ γ/K_L) separation with the ECAL alone
  - (Beyond paper) Our gains on π^0 photon pre-clustering performance using Graph Theory suggest calo-PID could have a bigger role in PFA

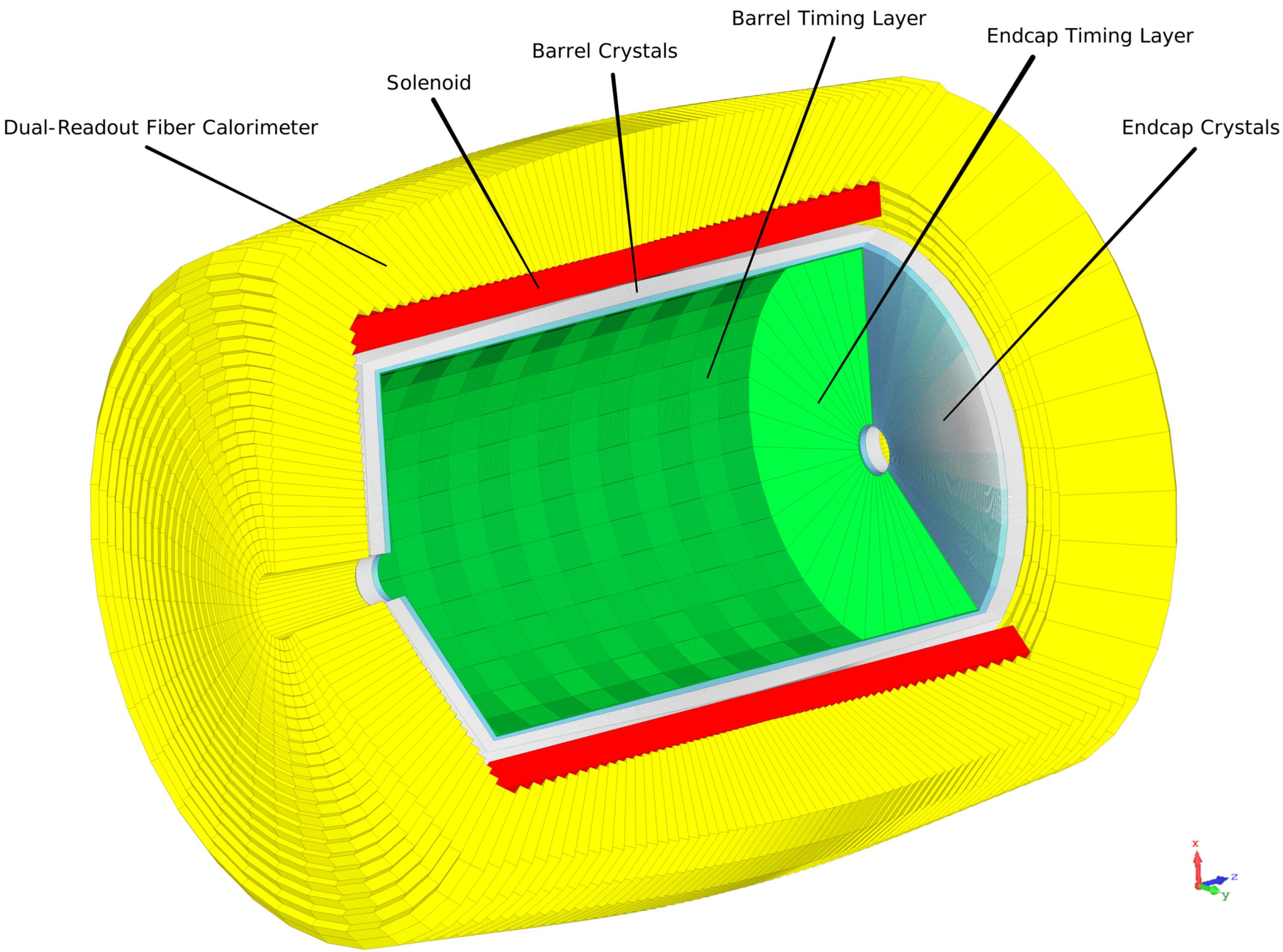
Segmented Crystal Option of IDEA



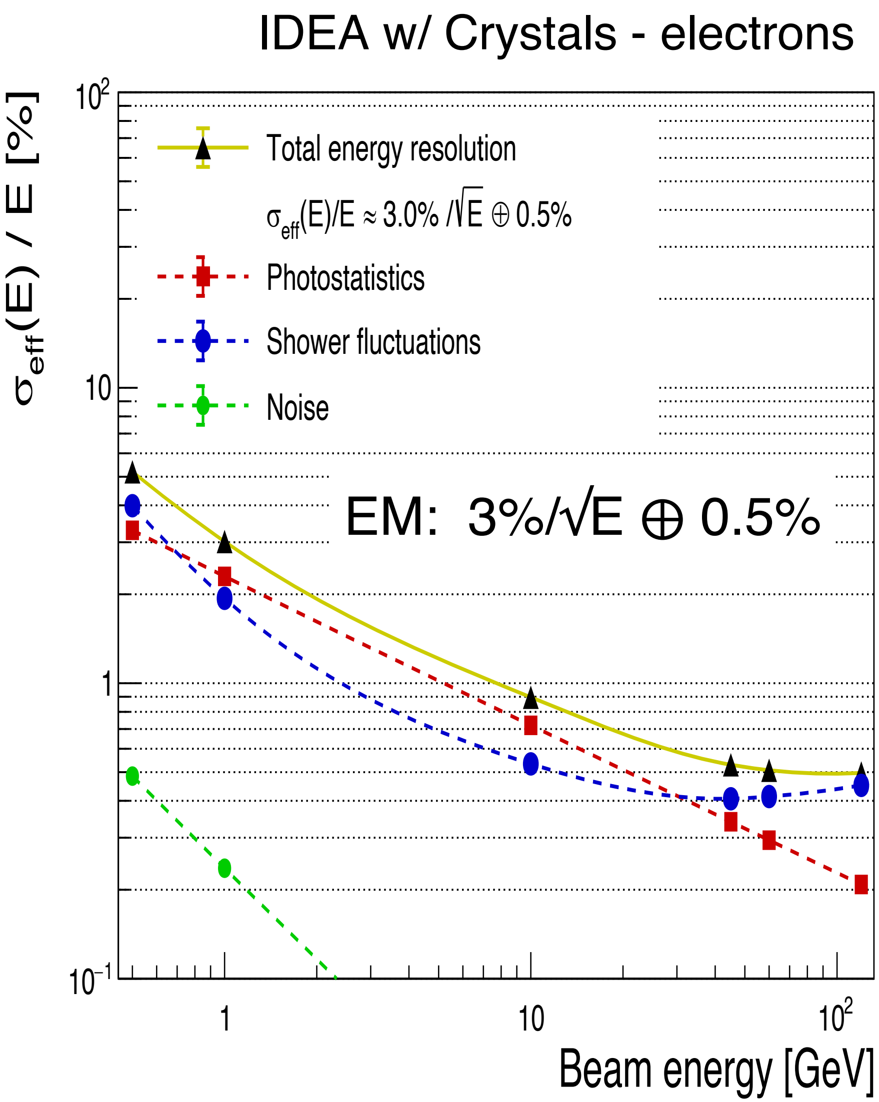
Front/Rear Crystal Transverse Segmentation

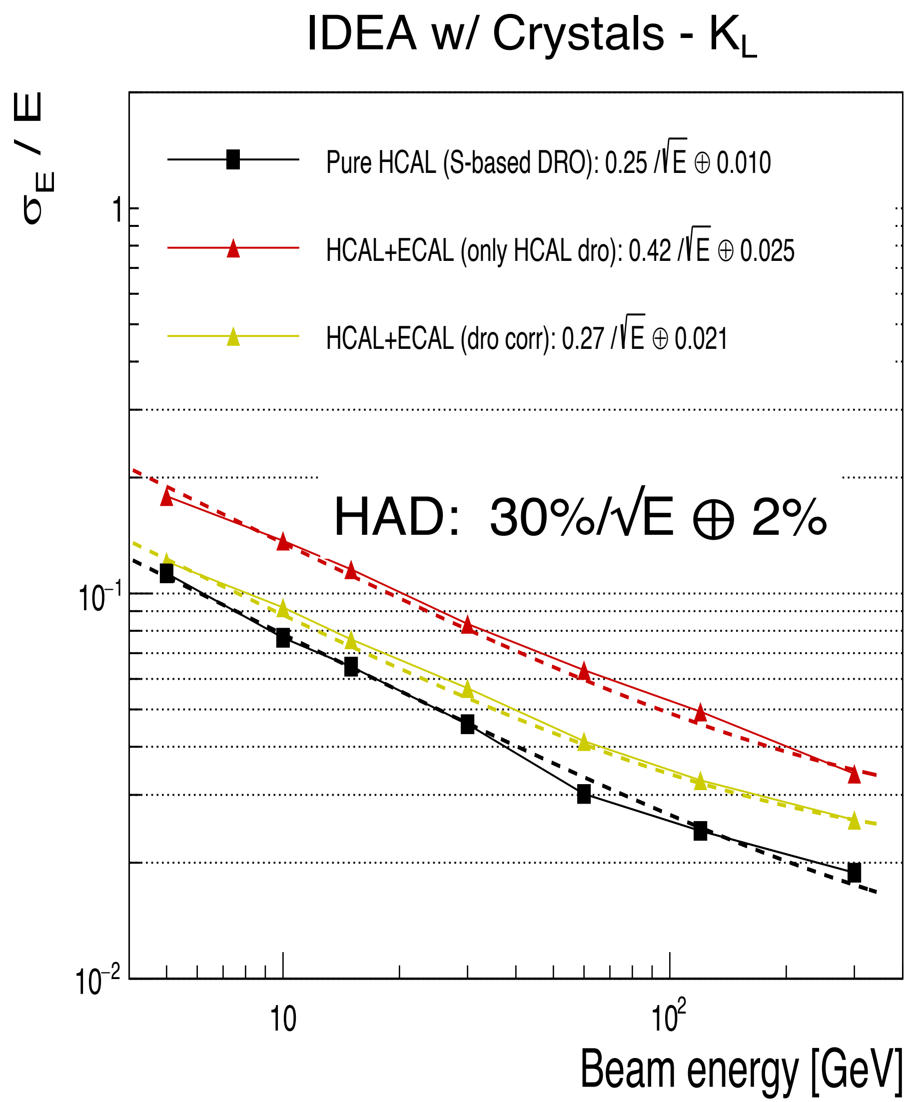


Full Geometry Implementation



Photon and Neutral Hadron Energy Resolutions

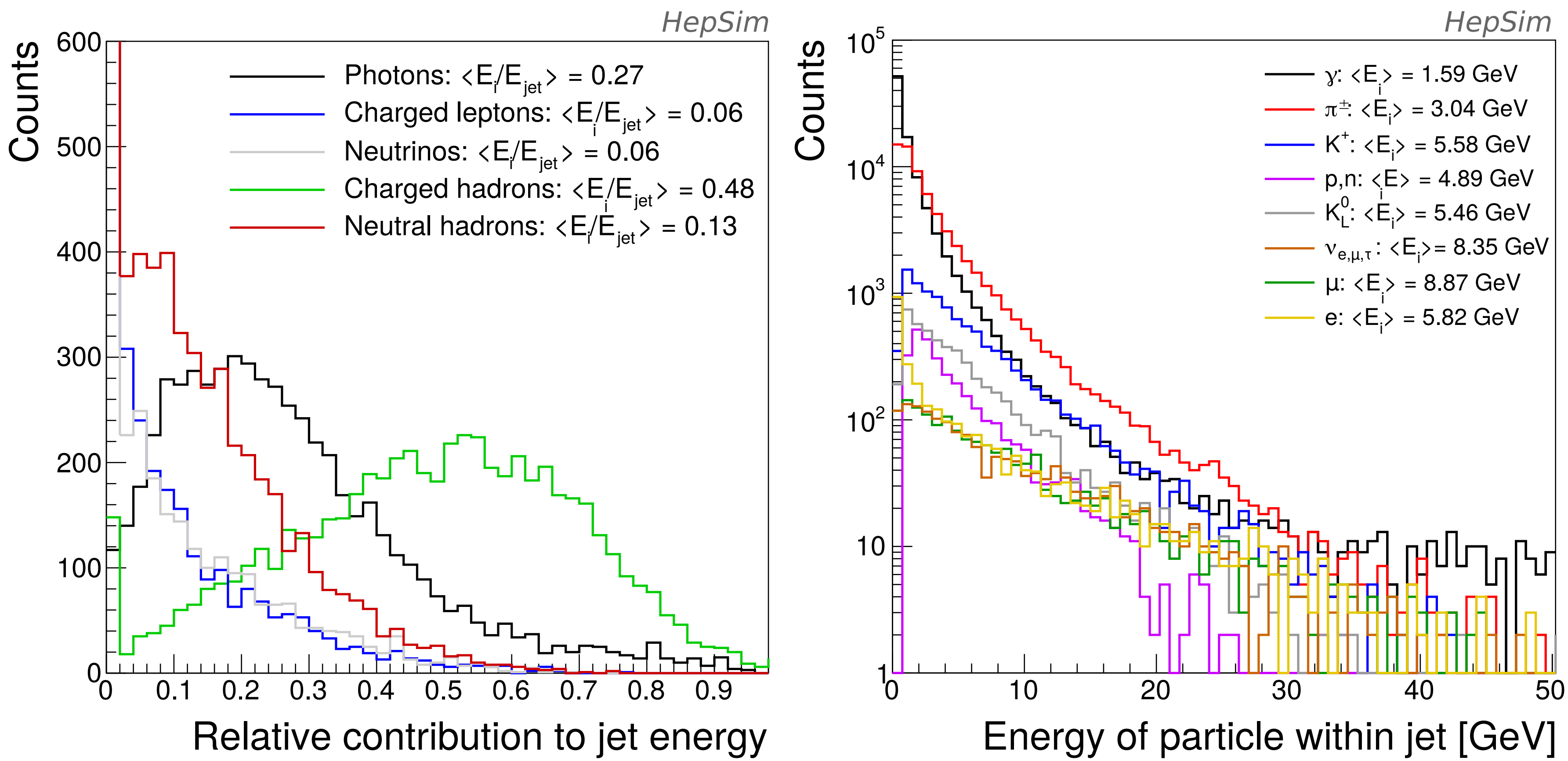




Energy Resolution Target: <3%/√E Stochastic Term

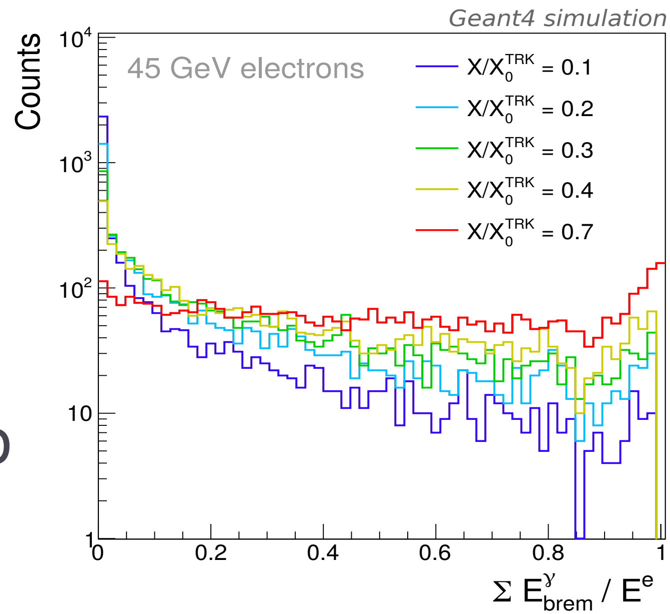
- **Requires:**
  - Shower fluctuations <2%
    - Material budget in front of ECAL < $0.3X_0$
  - Photostatistic fluctuations < 2%
    - Signal in photoelectrons >3500 phe/GeV
    - Assuming 20% PDE for 10 um cell SiPMs → LY*LCE > 18 ph/MeV
      - 5 um cells with high PDE in development
    - Need to tune SiPM active area accordingly to crystal LY
      - PWO: LY=100 ph/MeV → LCE>18% → SiPM area > 64 mm²
        - SiPM number of cells: 360k
      - BGO: LY=7000 ph/MeV → LCE>0.3% → SiPM area > 1 mm²
        - SiPM number of cells: 10k → dynamic range effectively x30-40 larger due to the fast time response of the pixel compared to the BGO decay time

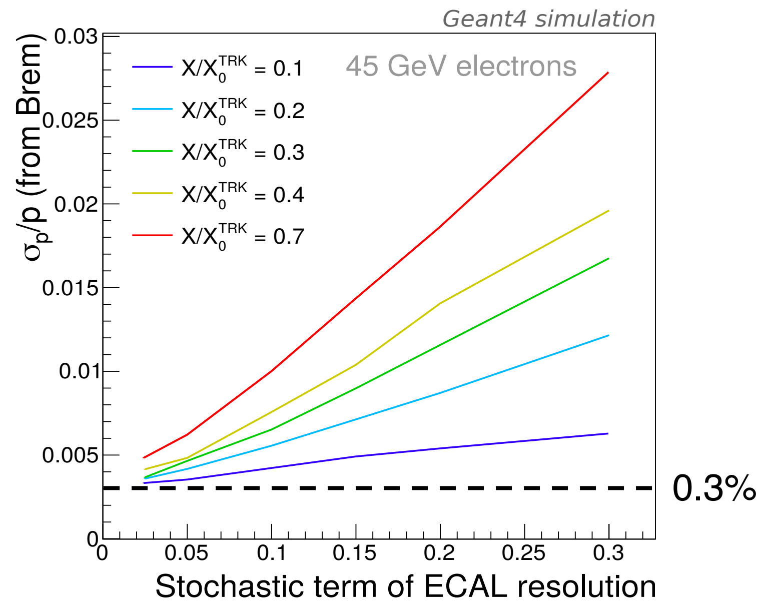
Neutral Hadron Spectra and Jet E Fraction

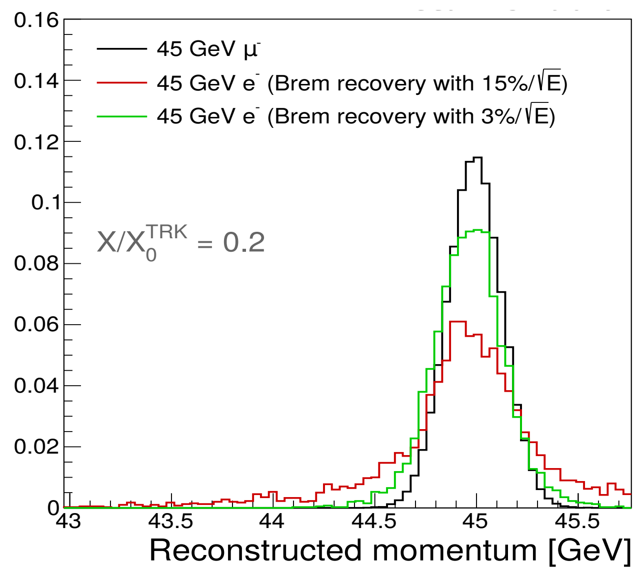


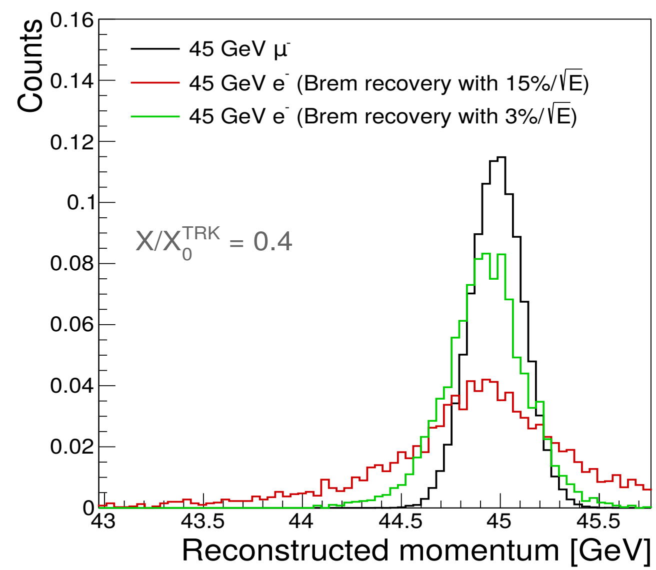
Z→ee Bremsstrahlung Recovery

Assuming tracker low-p (multiple scattering) resolution $\sigma_p/p = 0.3\%$

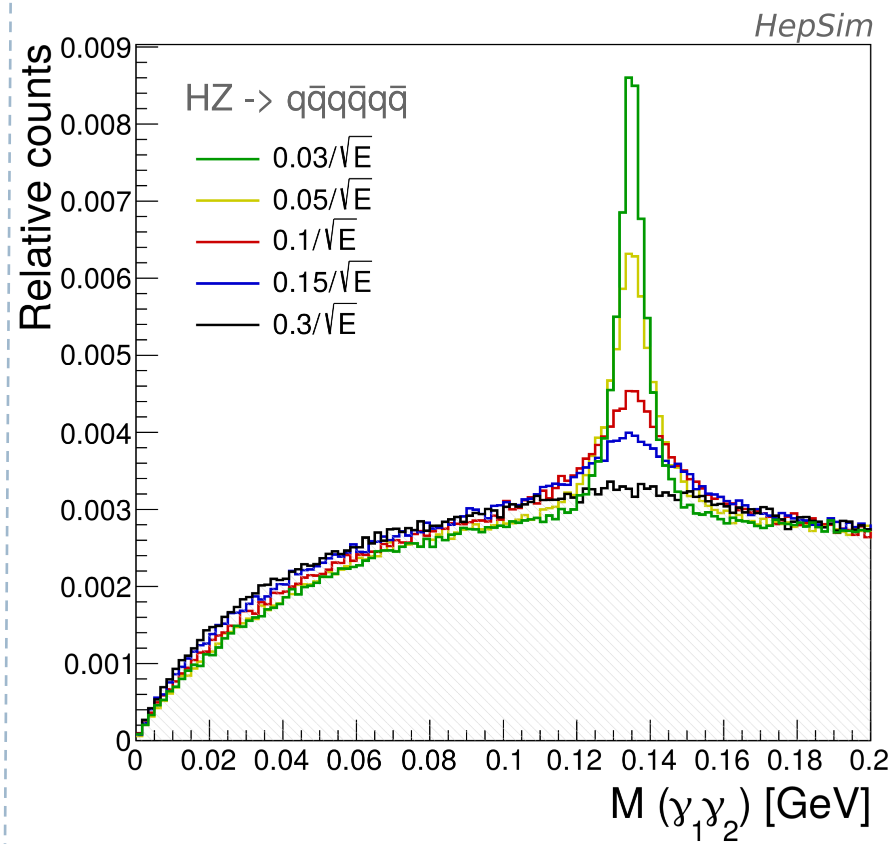


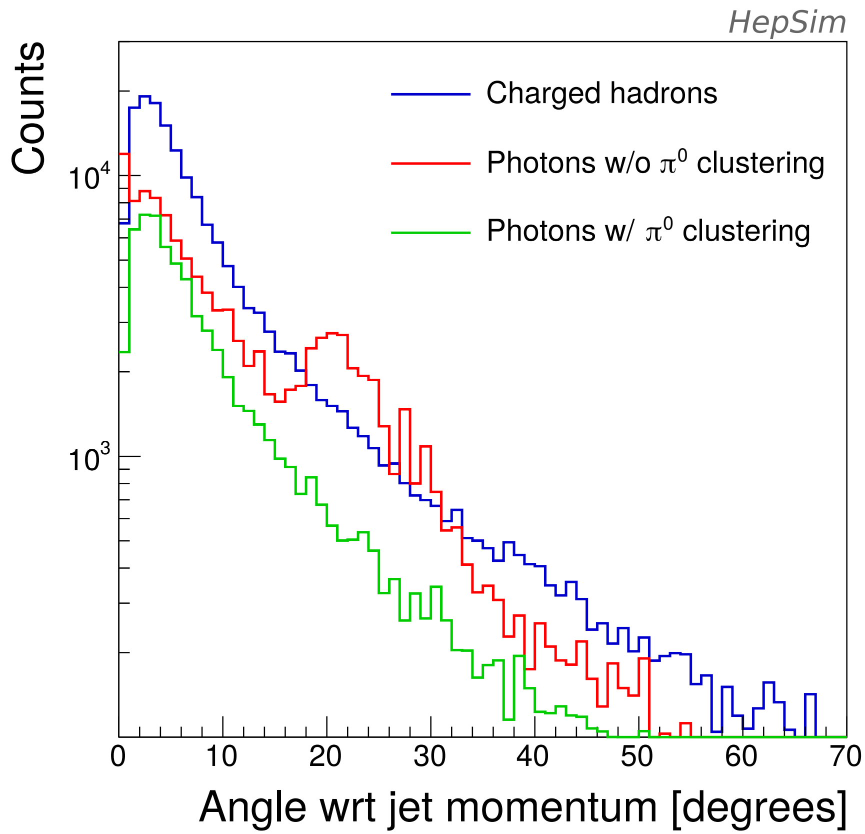




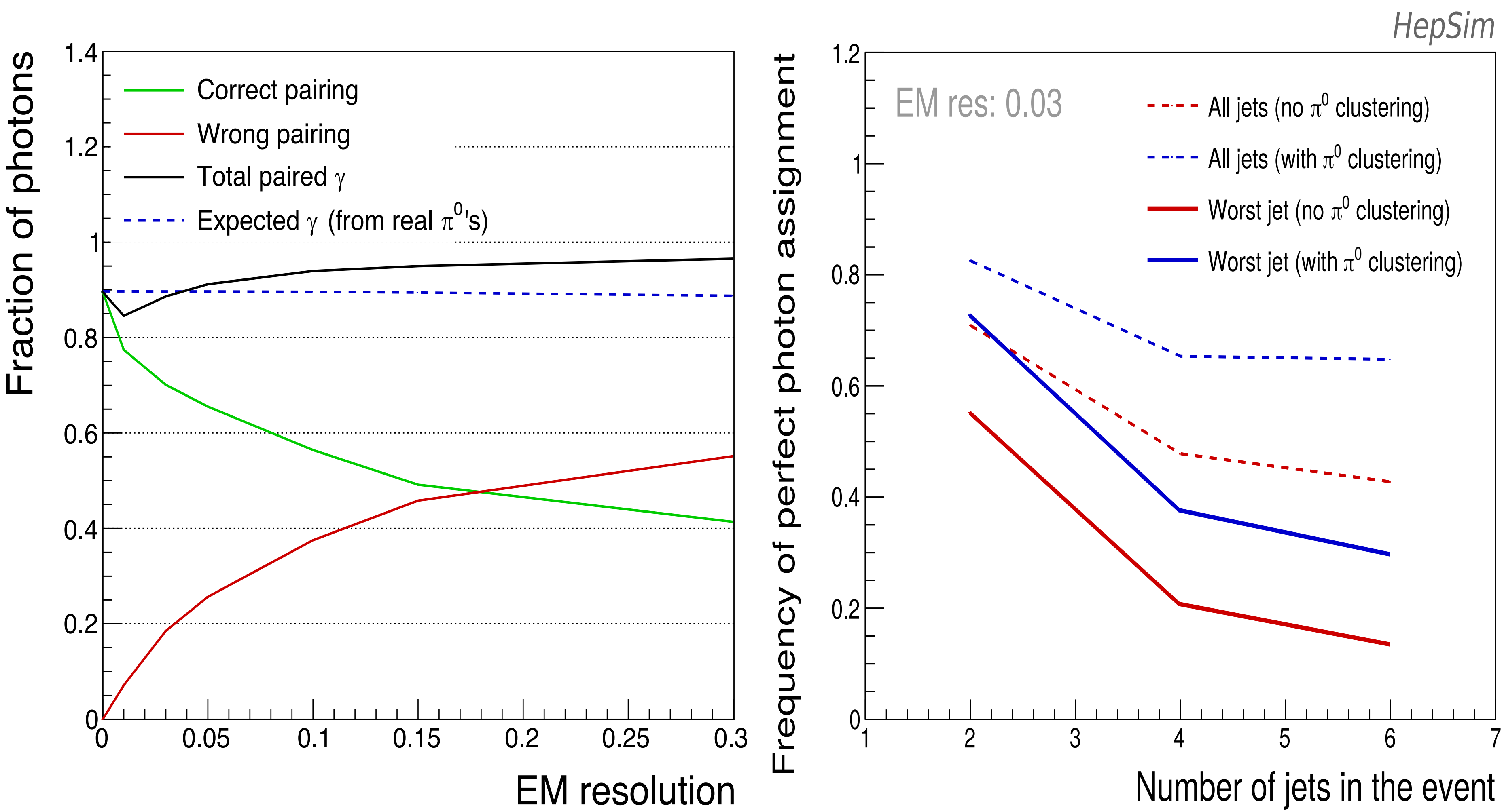


Pre-Clustering of π^0 Photons



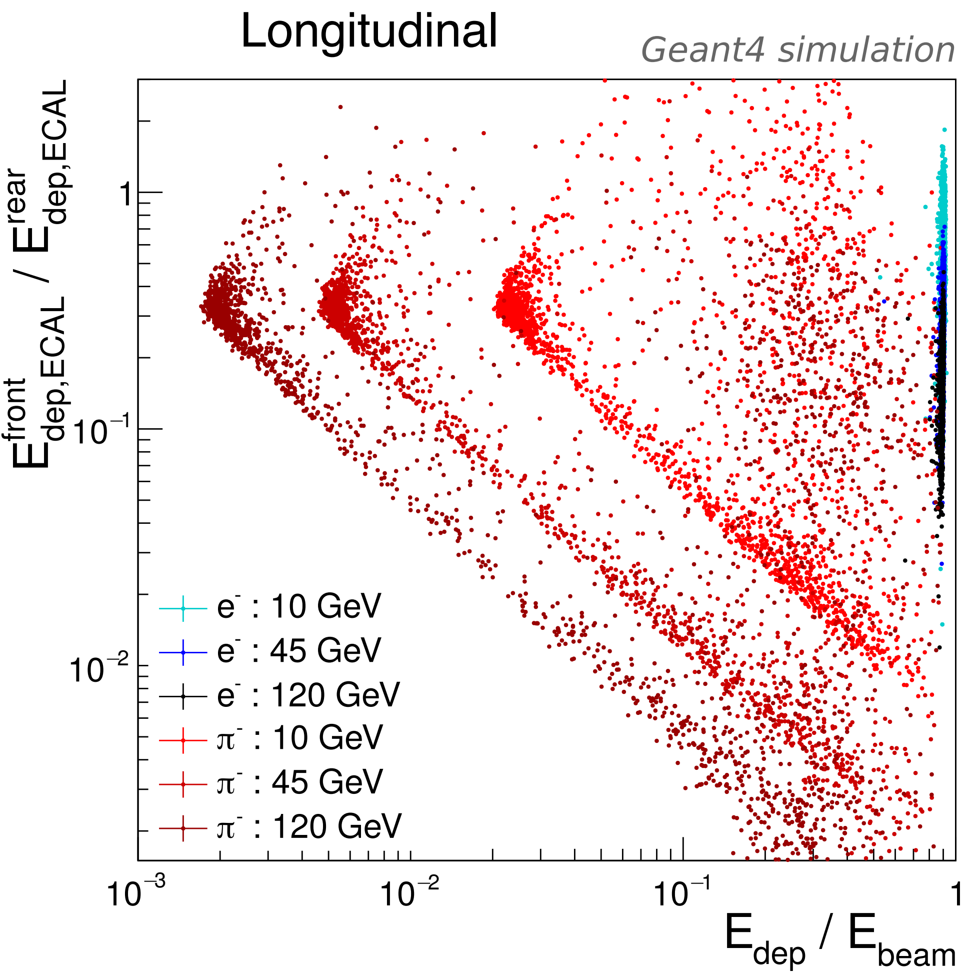


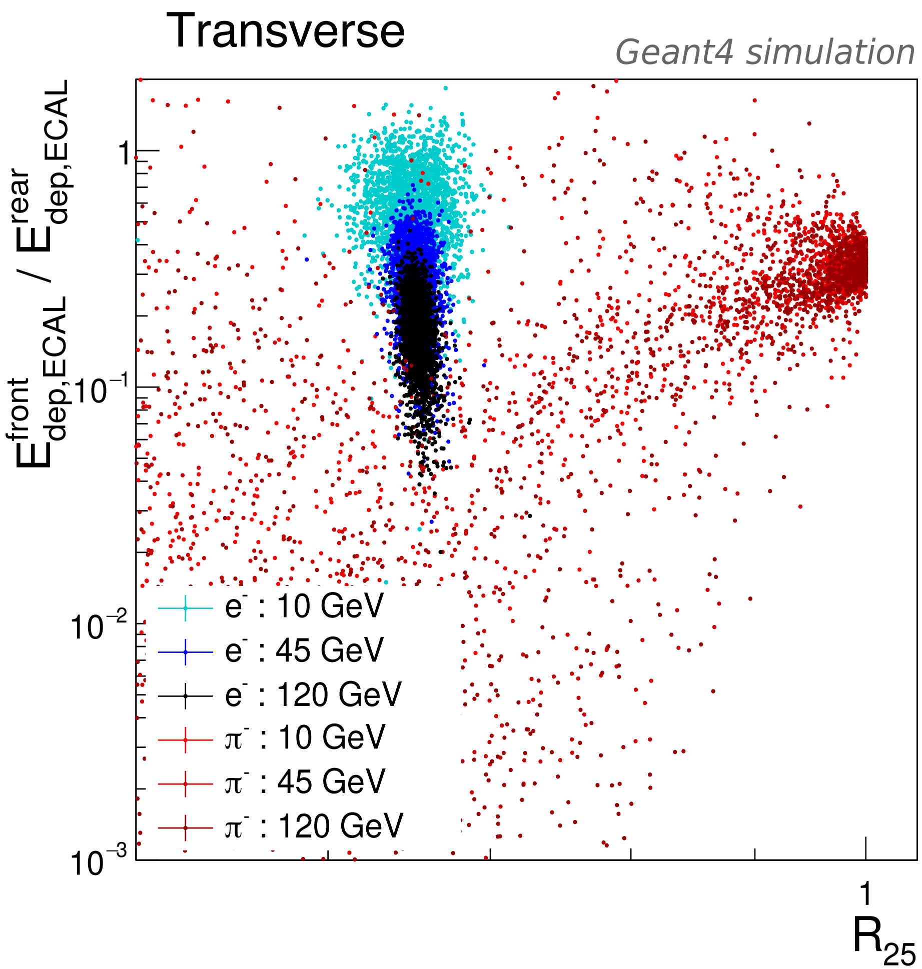
Perfect π^0–to–π^0 Photon Jet Correspondence



ECAL-Only Particle Identification

Longitudinal+Transverse: 99% Electron Efficiency @ 99% Pion Rejection

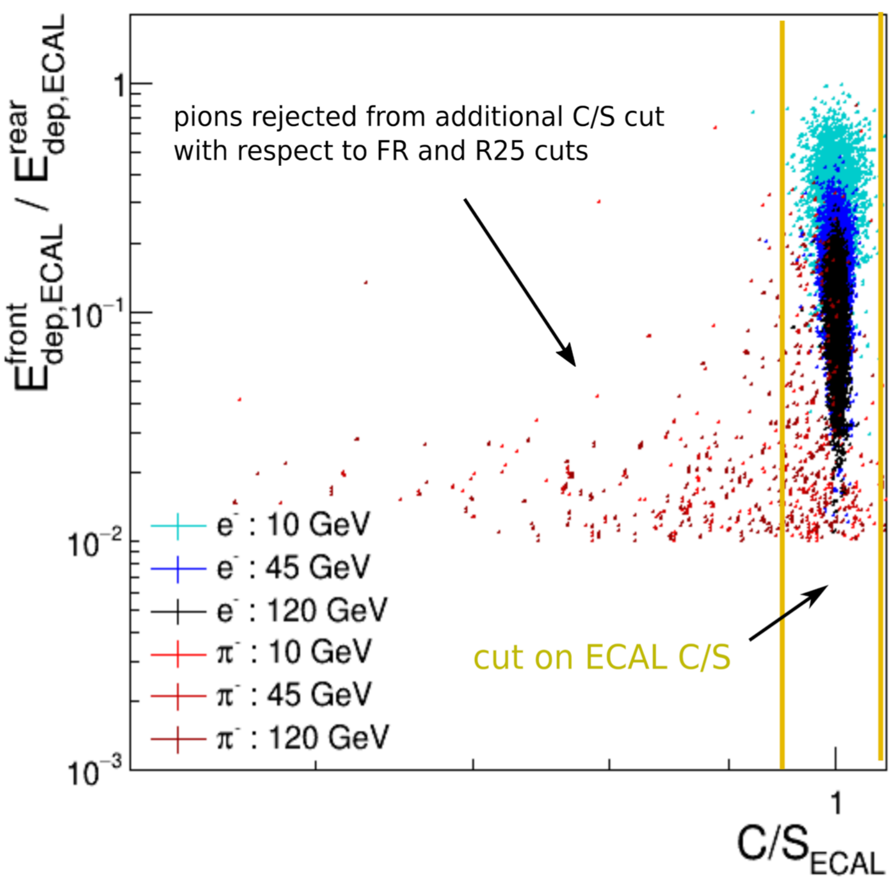


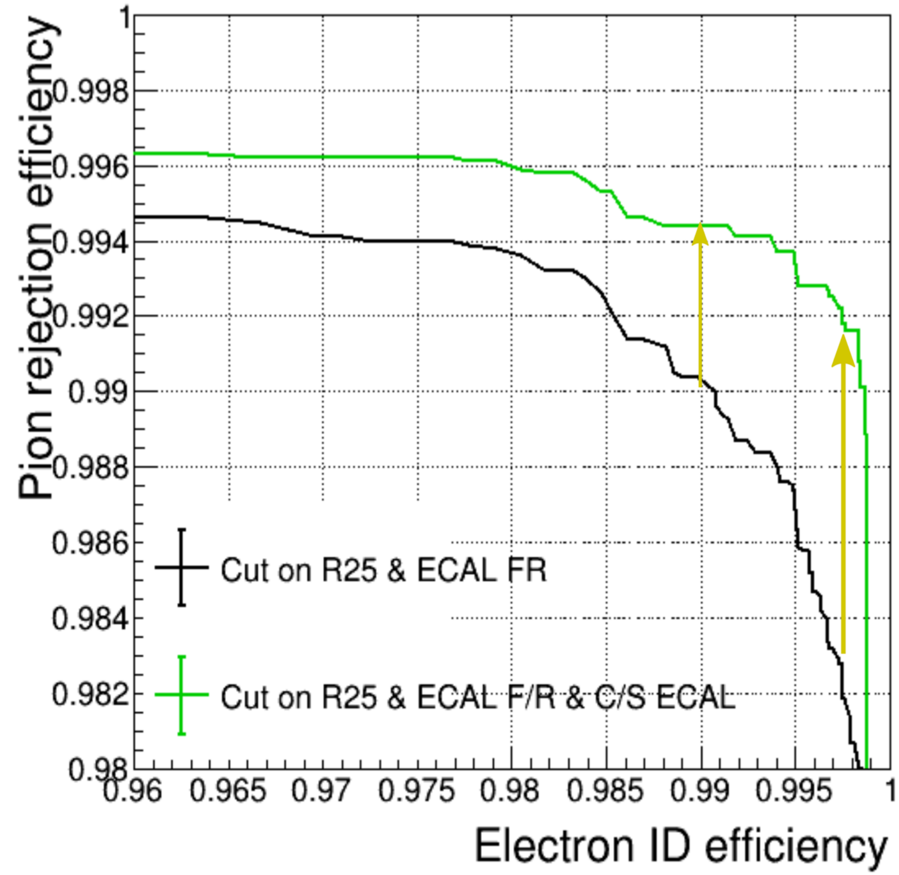


More ECAL-PID Possible

<u>99% Electron Efficiency @ 99.4% Pion Rejection</u>

Submitted to JINST Referee Report





+ PID from Precision Timing Layer

Graph Theory Applied to π^0 Photon Pre-Clustering

Algorithm

- max_weight_matching (*G*, *maxcardinality=False*, *weight='weight'*)
- Compute a maximum-weighted matching of G.
  - A matching is a subset of edges in which no node occurs more than once.
  - The weight of a matching is the sum of the weights of its edges.
  - A maximal matching cannot add more edges and still be a matching.
  - The cardinality of a matching is the number of matched edges.
- If G has edges with weight attributes the edge data are used as weight values else the weights are assumed to be 1.
- This function takes time O(number_of_nodes ** 3).

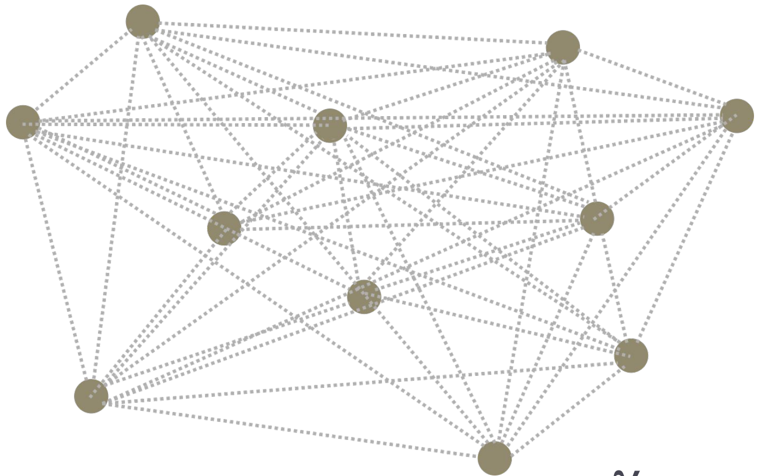
- This method is based on the “blossom” method for finding augmenting paths and the “primal-dual” method for finding a matching of maximum weight, both methods invented by Jack Edmonds [1].

[1] https://dl.acm.org/doi/10.1145/6462.6502
[max_weight_matching] https://networkx.github.io/documentation/stable/reference/algorithms/generated/networkx.algorithms.matching.max_weight_matching.html

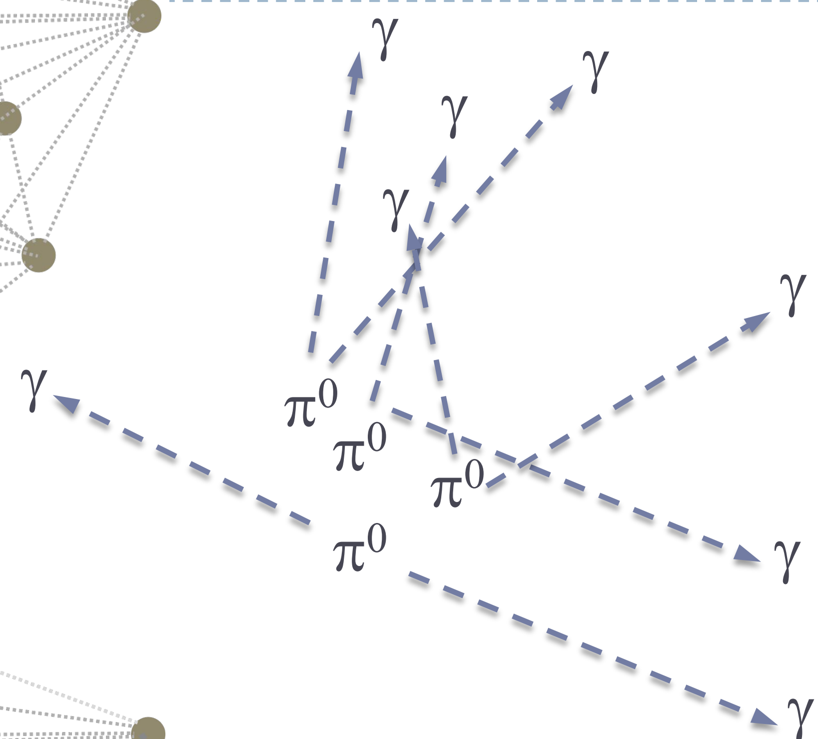
M./L. Lucchini

Building graph

- **node** = photon
- **edge** = pair of photons

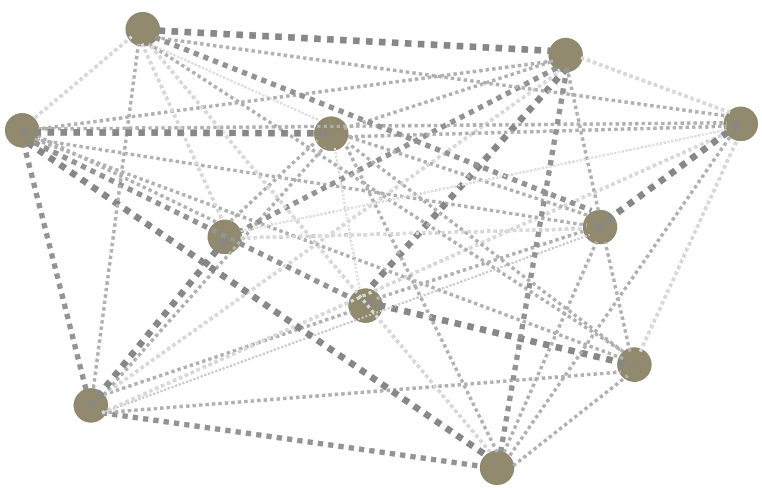


- node properties
  - px, py, pz, E
- edge properties
  - invariant mass
  - boost
  - angle



Assigning weights

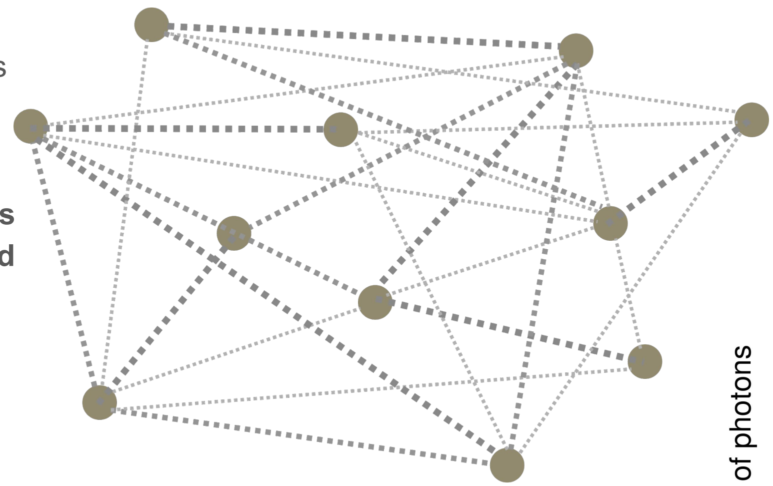
- Assign a **weight, w_{ij}**, to each edge
- $\chi^2_{ij} = (M_{\gamma,i\ \gamma,j} - M_\pi)^2 / M_\pi$
- $w_{ij} = 1 - \chi^2_{ij} / \chi^2_{max}$
- $\chi^2_{max} = \max(\chi^2_{ij})$
- $w_{ij} \in [0,1]$

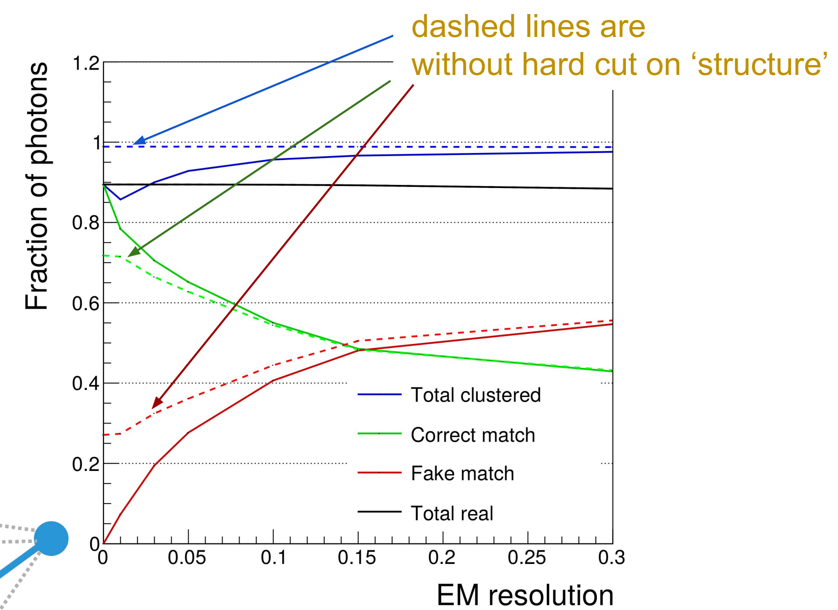


M./L. Lucchini

Defining underlying structure

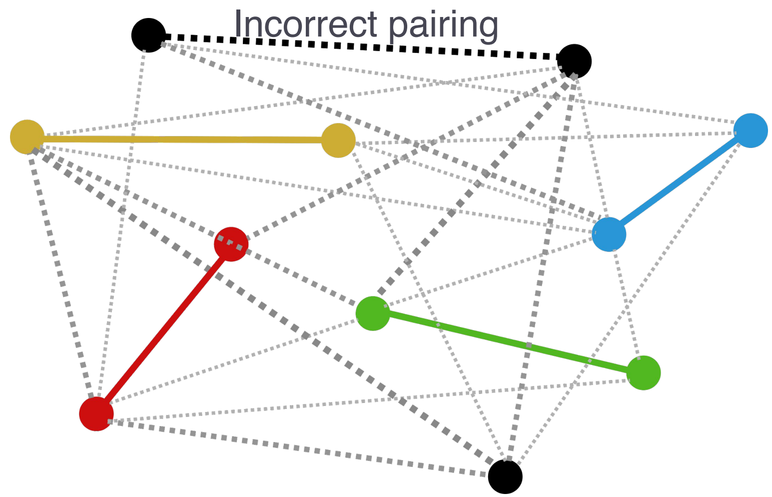
- Define a **structure** of the graph that describes which edges can be accepted
- In practice, **reject edges that do not pass a hard cut** on certains variables, e.g.:
  - $|M_{\gamma\gamma} - M_{\pi}| > M_{\pi} * 3\sigma_{EM}$
  - boost
  - angle





Finding solution

- Use each node at most once
- Maximize: $\mathbf{\Sigma w_{ij}}$
- Return set of photon pairs



M./L. Lucchini

Thinking Ahead

- **Improvement from Graph Theory on pre-clustering of π^0 photons was substantial**
  - Keeps tabs on bad pairings – not enough to have the “best” pairing for a given photon
- **With an ECAL that can self-select photon or neutral hadron using PID, pairings of EM/HAD clusters and pairings of track/EM/HAD that keep track of bad/inconsistent PID may allow for better PFA outcomes**
  - Optimizing a high resolution EM/HAD calorimeter for PID-matching could provide a more precise event description, higher identification rates for rare processes and lower overall systematic uncertainties

Some Initial Ideas on Graph-PFA

- **1st Order PFA :**
  - swaps out hadron showers with tracks
  - photons are separated out with ECAL
  - electrons/muons separated out by PID
  - neutral hadrons are "leftovers"

- **2nd Order PFA: ?**
  - PID per cluster creates weight assigned to a potential EM/HAD/track match
  - created graphs of all possible pairings (subject to structure constraints/1st Order PFA), assign weights, graphs with unmatched tracks/inconsistent PID/isolated cluster fragments have low overall weight
  - Highest weight graph may improve event description and provide a new benchmark for global detector performance

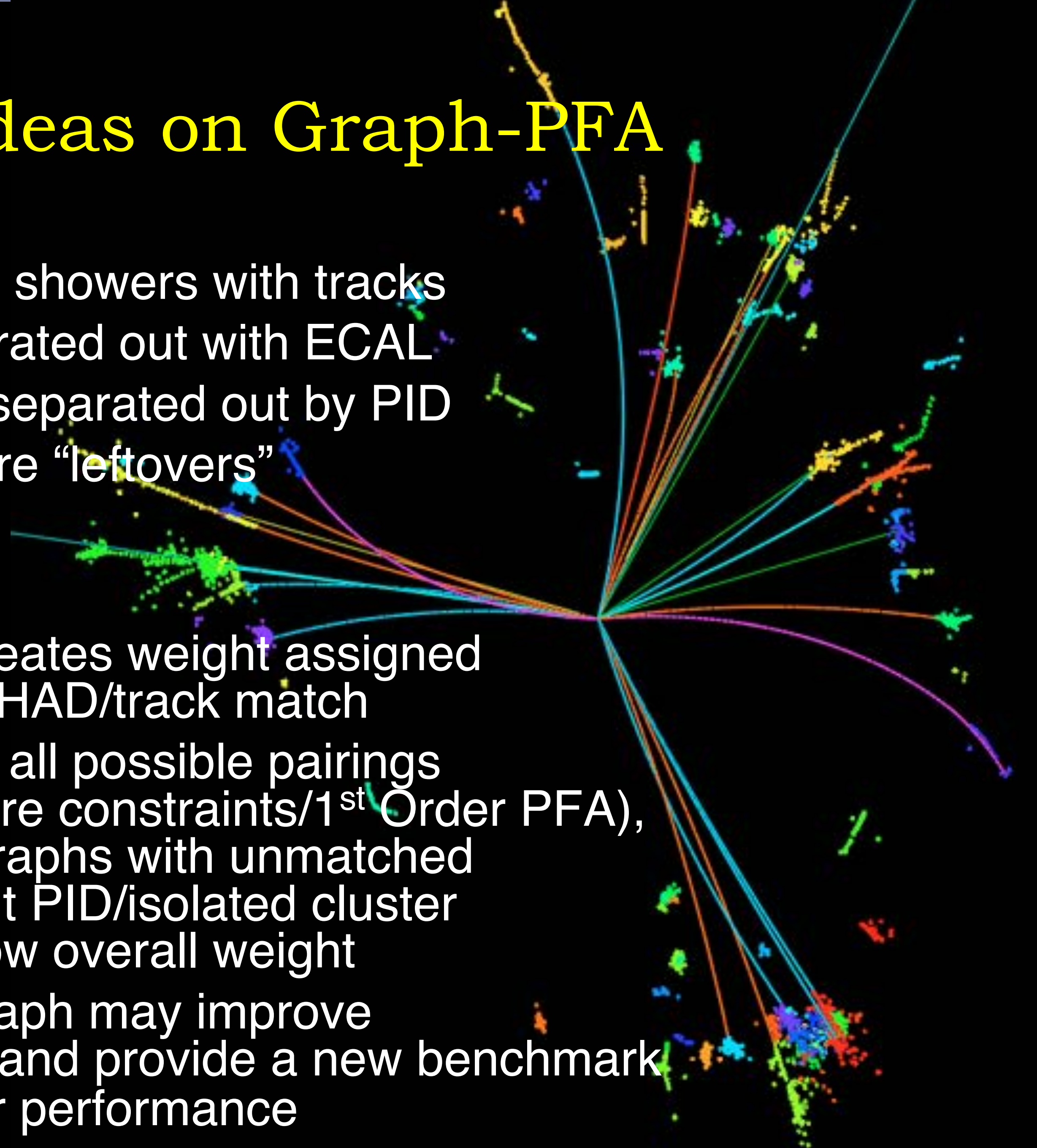
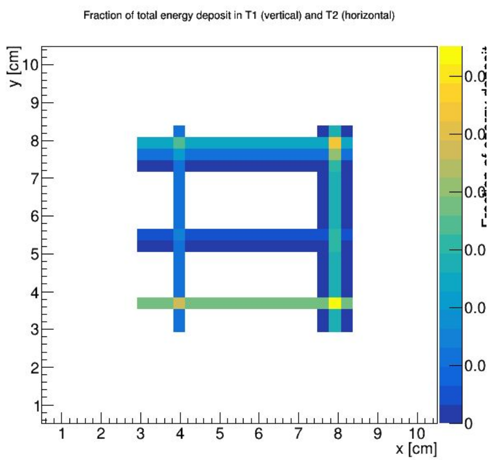
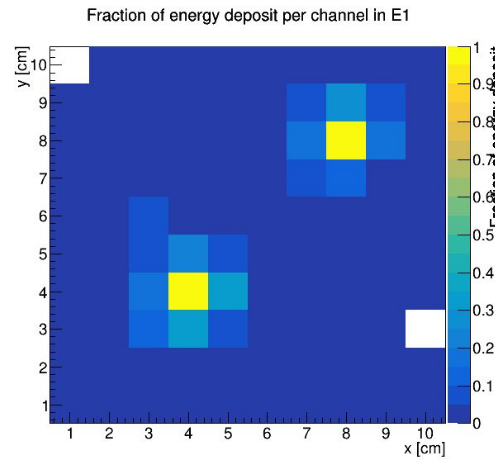


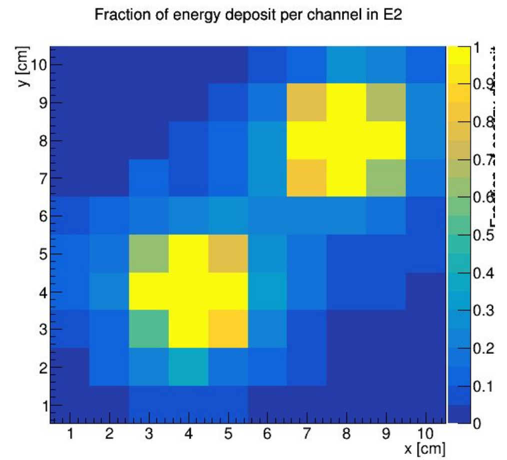
Image from Manqi Ruan

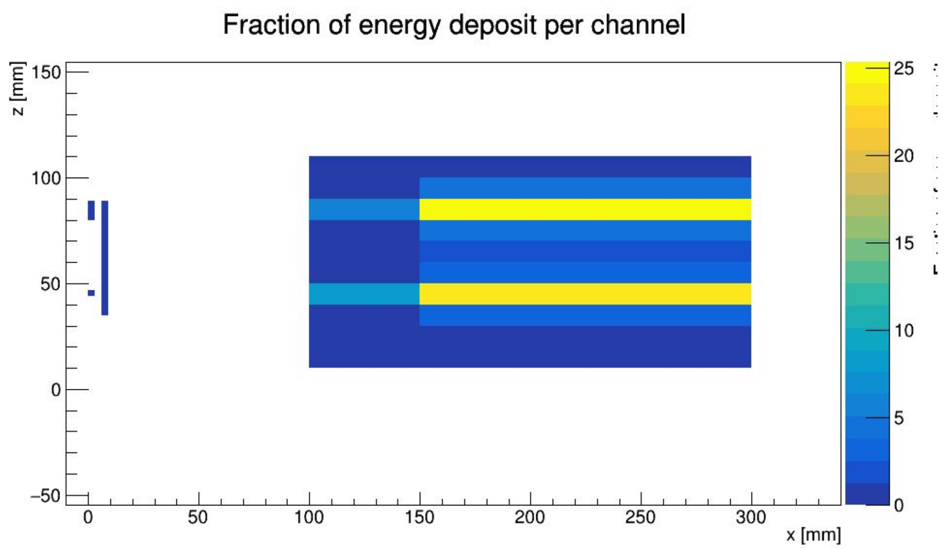
Additional slides

Pair of EM Showers (Single Event)

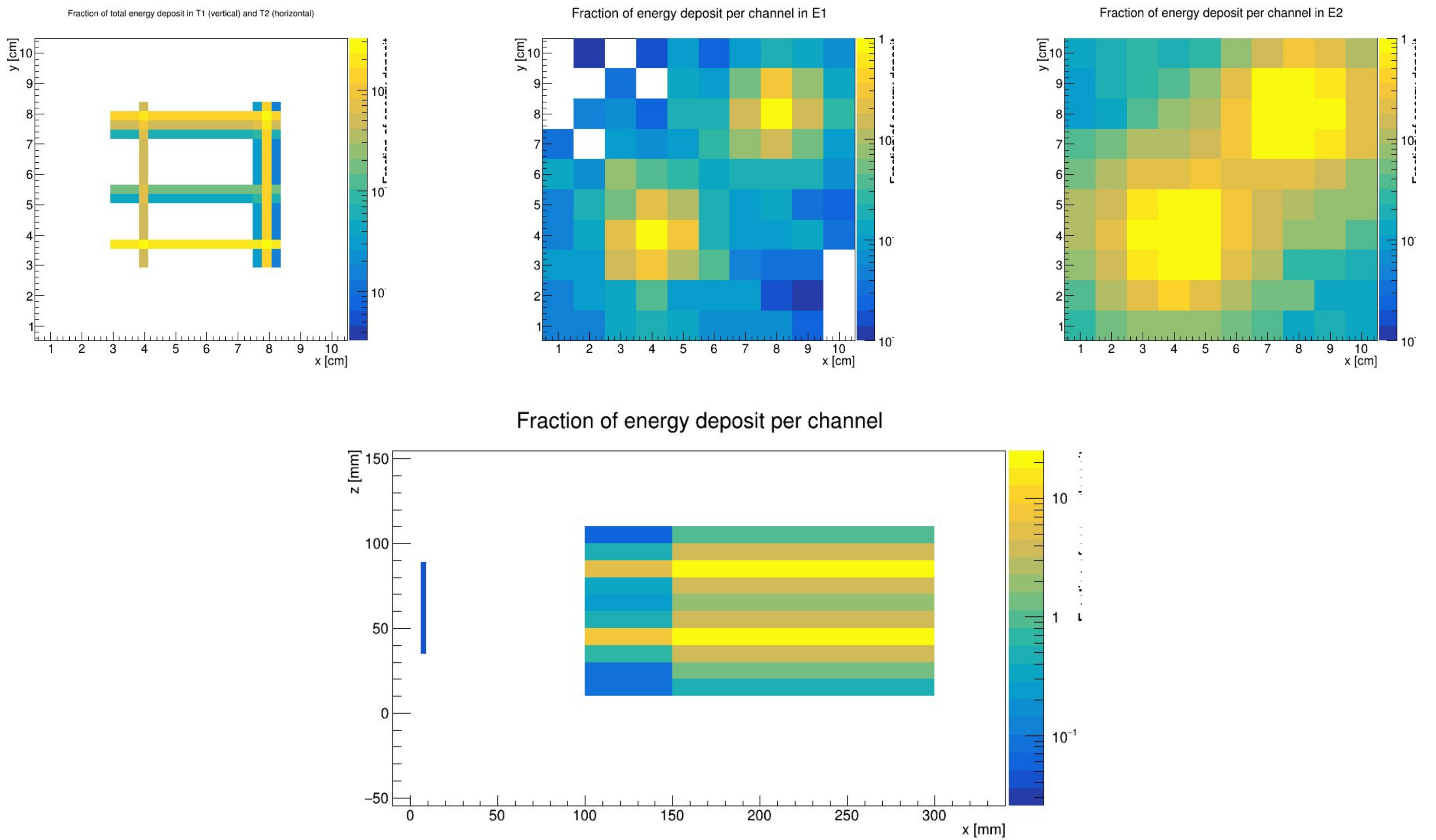






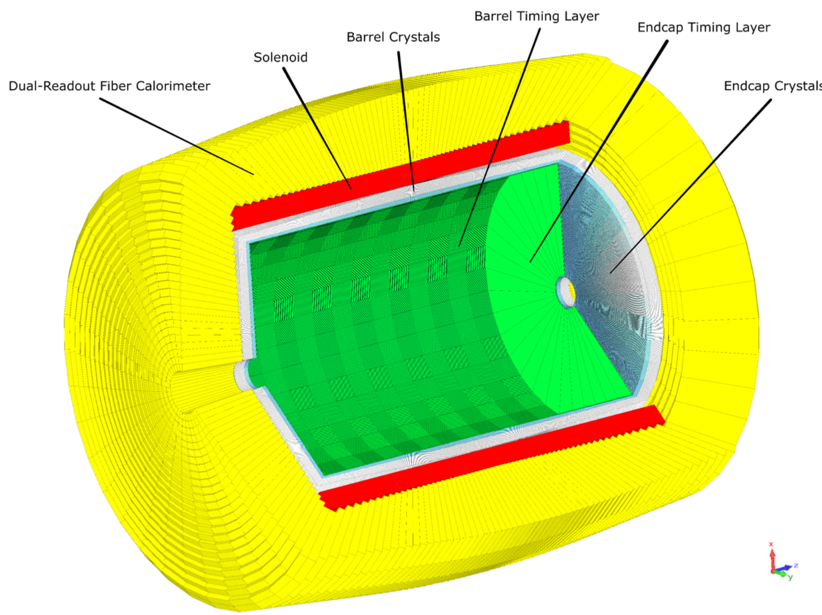


Pair of EM Showers (Single Event - Log)

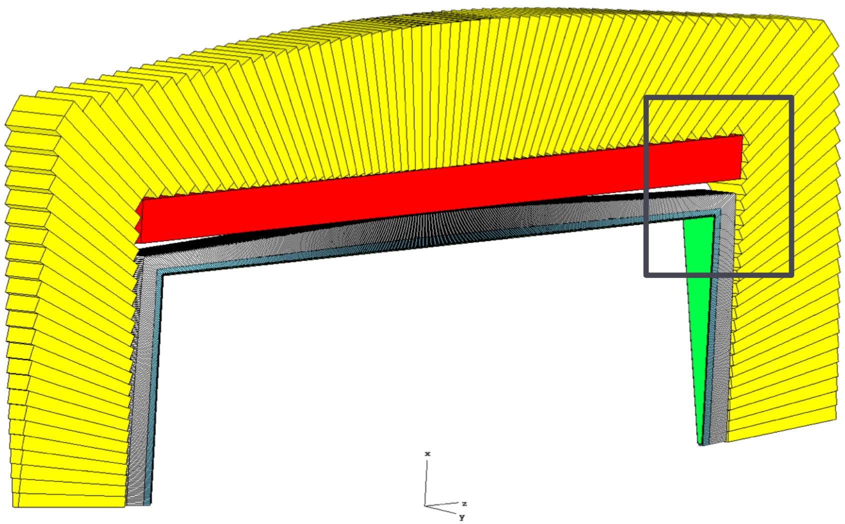


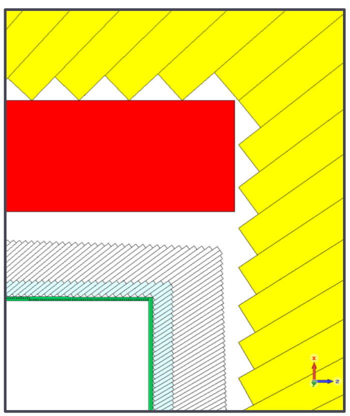
Detector Geometry Guide

Wonyong Chung (wonyongc@princeton.edu)



Any questions, please don't hesitate to email me!





Detector Geometry General Overview

Components:

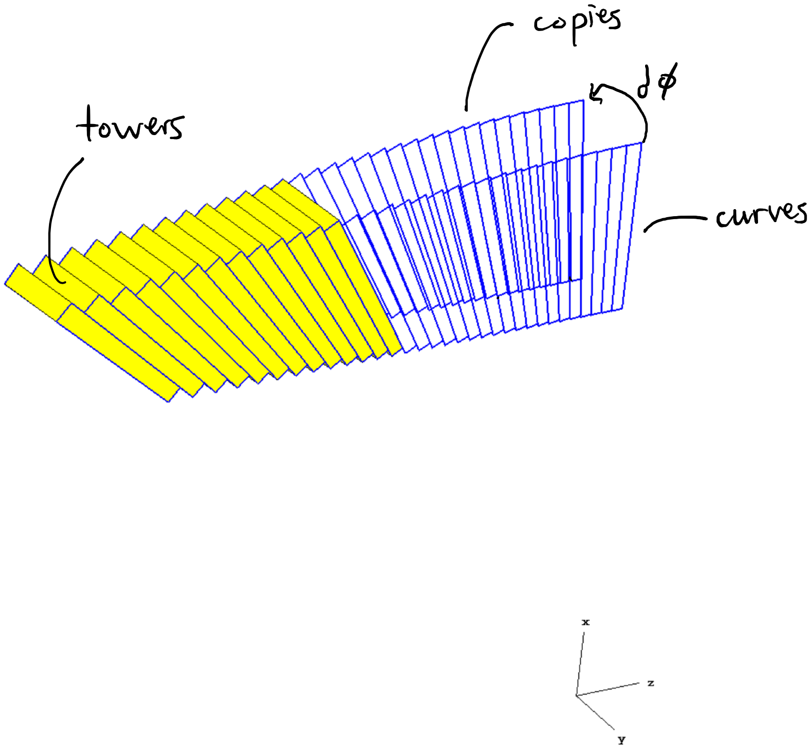
Solenoid (red)
HCAL (yellow)
ECAL (white/blue)
Timing Layer (green)

One phi slice of each component is made in each ‘MakeComponentName’ function.

The function ExecuteRotationsInPhi then copies each component into the desired number of rotations in phi.

Dimensions of the ECAL are specified first and most others are derived from them.

Some more details in the comments at the top of the script file



This document specifically details the geometry calculations for the HCAL and ECAL towers.

Outlines of the z-x cross sections of each tower are defined (curves) and one copy of each curve is made at an angle dphi from the original. Solids (towers) are generated by filling in the volume between these two curves in the shortest distance (straight line).

For the ECAL crystals, additional cuts are made in phi for each tower to produce individual crystals.

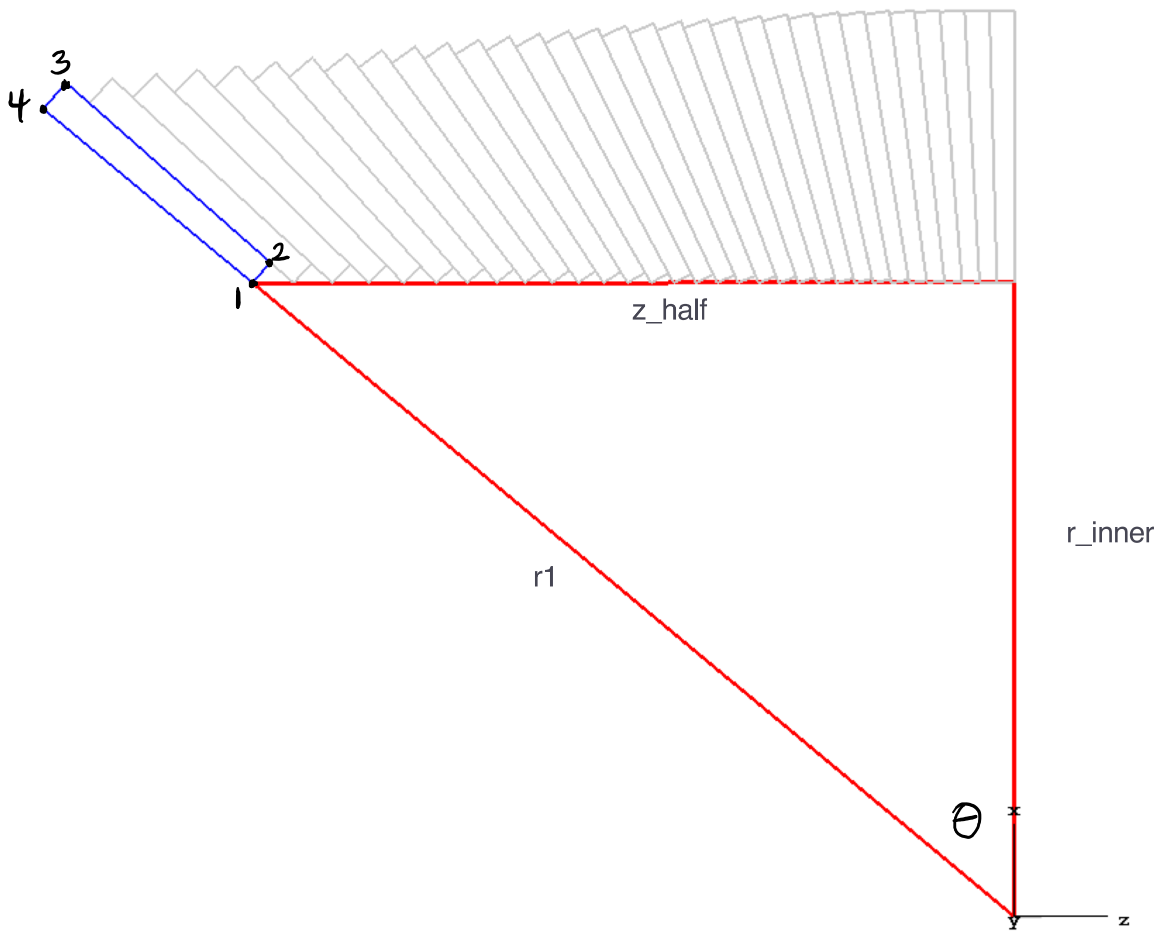
For the tower curves, z and x coordinates are calculated to make a closed planar polygon in the z-x plane.

The calculations take the provided dimensions and otherwise make two fundamental assumptions:

1. Each tower reads an equal slice of theta from the interaction point. Therefore dtheta is constant for each tower and the dimensions of individual tower faces will differ slightly. The nominal_tower_face dimension is therefore just an estimate.

2. Each tower face makes a 90 degree angle with the line connecting the midpoint of the tower face and the interaction point. Therefore each tower face is the base of an isoceles triangle with the interaction point as the apex.

The variable s is used throughout to denote the sign of the z coordinate.



Starting from the outermost tower (farthest in theta).

Theta extends from x axis.

theta = arctan(z_half/r_inner)

n_towers =
Floor(z_half/nominal_tower_face)
dtheta = theta/n_towers

We can now calculate coordinates for points 1, 2, 3, 4 of the tower cross-section.

r_inner, theta, dtheta (called dth) known

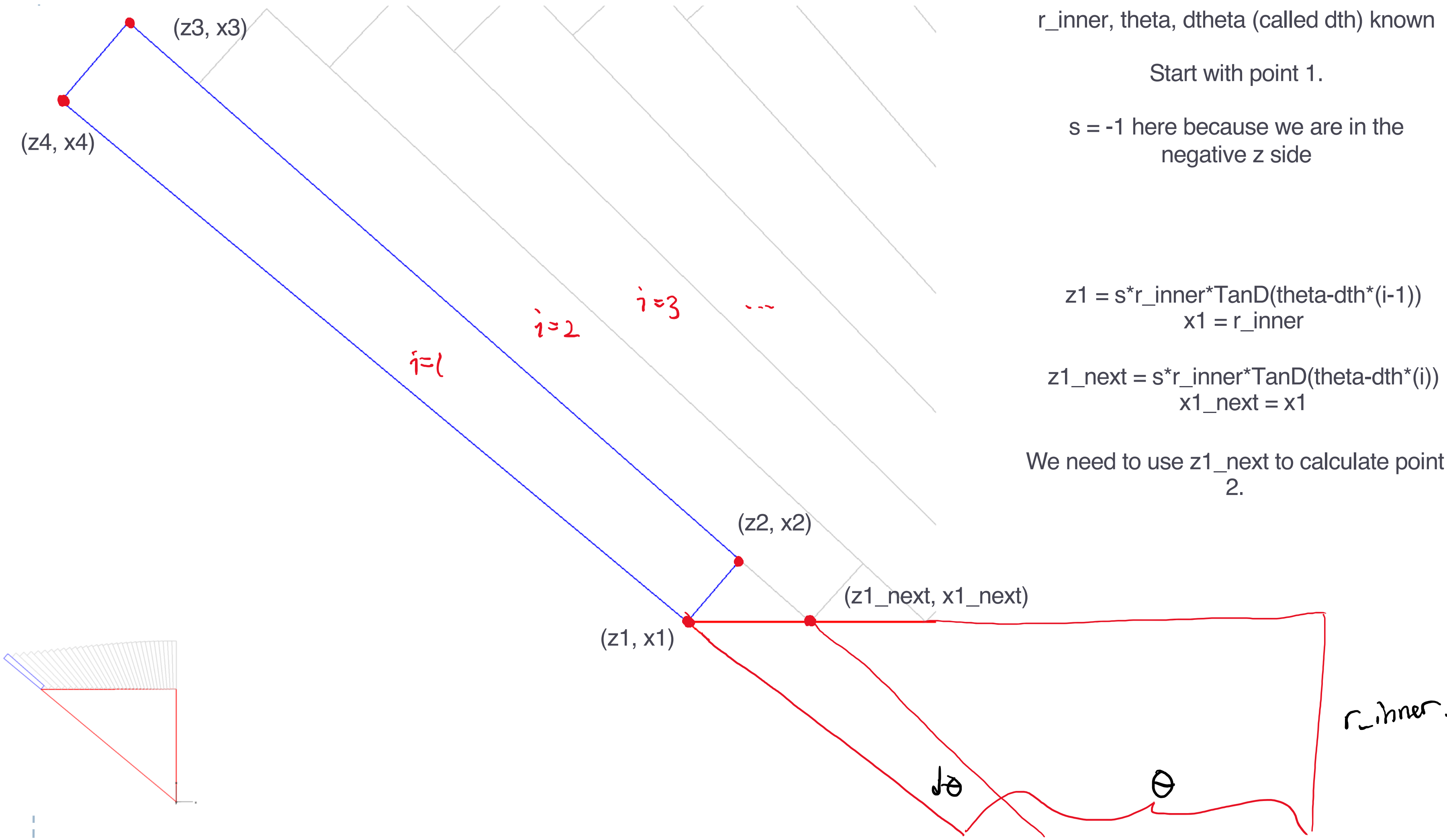
Start with point 1.

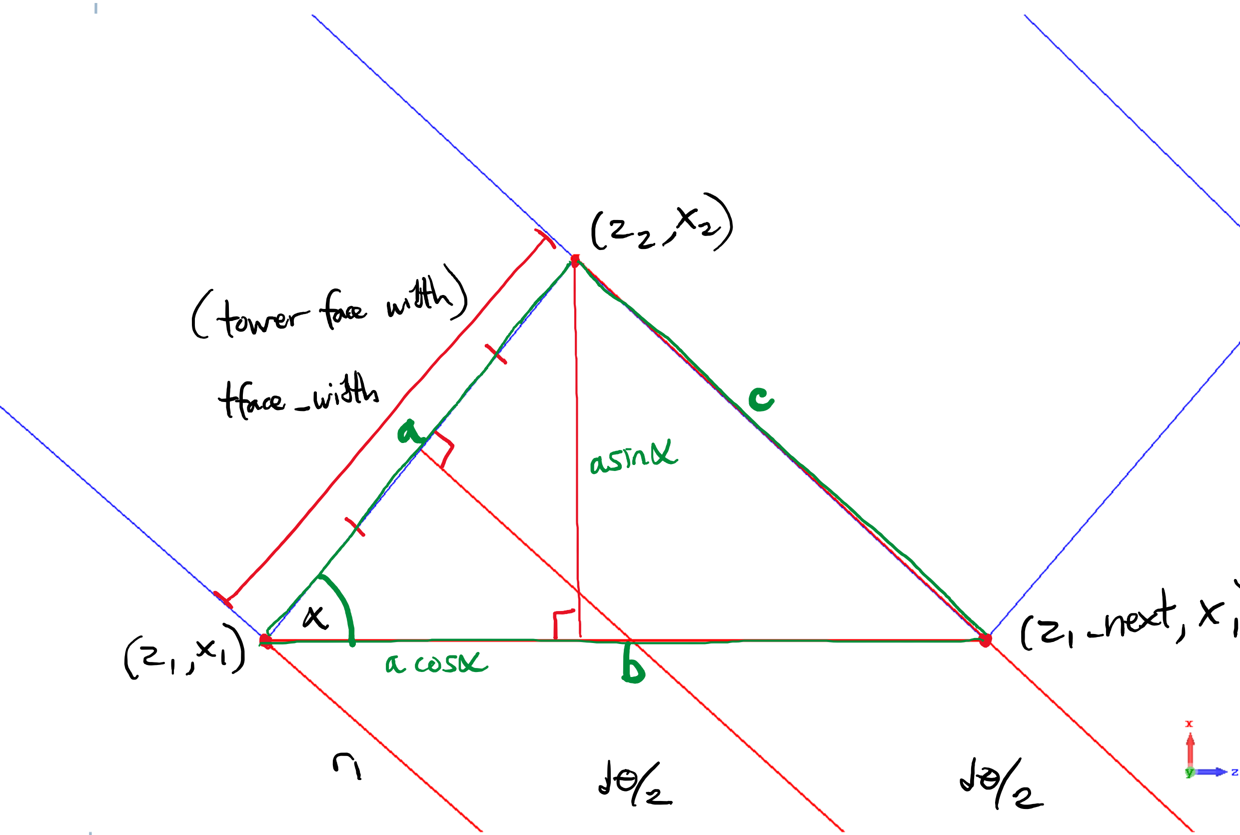
s = -1 here because we are in the negative z side

z1 = s*r_inner*TanD(theta-dth*(i-1))
x1 = r_inner

z1_next = s*r_inner*TanD(theta-dth*(i))
x1_next = x1

We need to use z1_next to calculate point 2.





z1, x1, dth known

=> r1 = Sqrt(z1^2+x1^2)

Calculate point 2 using angle alpha and tower face width (tface_width)

$$\rightarrow \frac{tface\_width}{2} = r_1 \cdot \sin(d\theta/2)$$

Law of Cosines to find α:

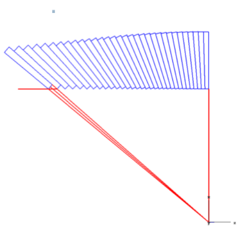
a = tface_width

$b = |z_1 - z_1 next|$

$c = r_1 - \sqrt{z_1 next^2 + x_1^2}$

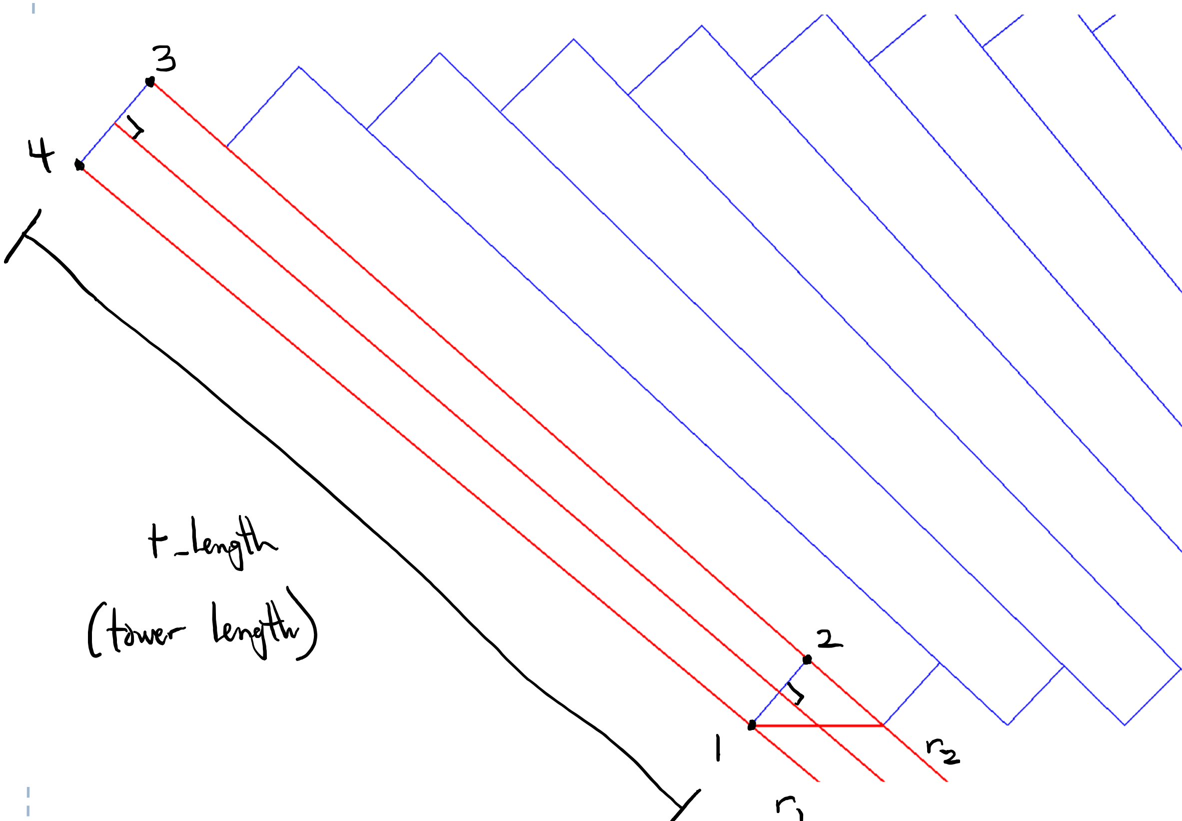
$c^2 = a^2 + b^2 - 2ab\cos\alpha.$

$\rightarrow \alpha = \arccos\left(c^2 - a^2 - b^2 / (-2ab)\right)$



Result:
z2 = z1 - s*a*CosD(alpha)
x2 = x1 + a*SinD(alpha)

s = -1 here because we are in the negative z side



Points 3 & 4:

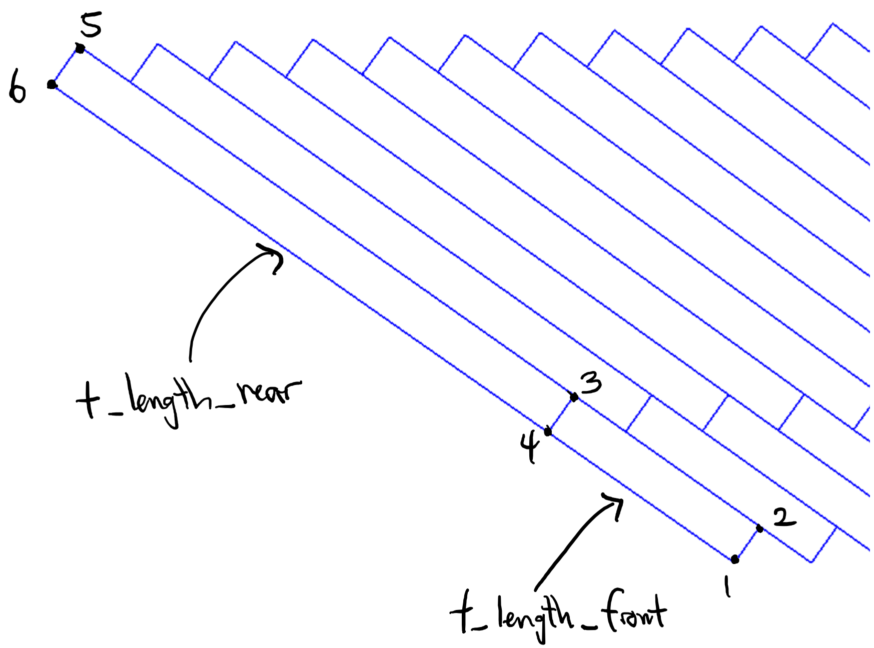
Extension of similar triangles from points 1 & 2.

$$r_1 = \sqrt{z_1^2 + x_1^2}$$

$$r_2 = \sqrt{z_2^2 + x_2^2} = r_1$$

$$\boxed{\begin{aligned} z_3 &= z_2 \cdot \left(\frac{r_2 + t\_length}{r_2}\right) \\ x_3 &= x_2 \cdot \left(\frac{r_2 + t\_length}{r_2}\right) \end{aligned}}$$

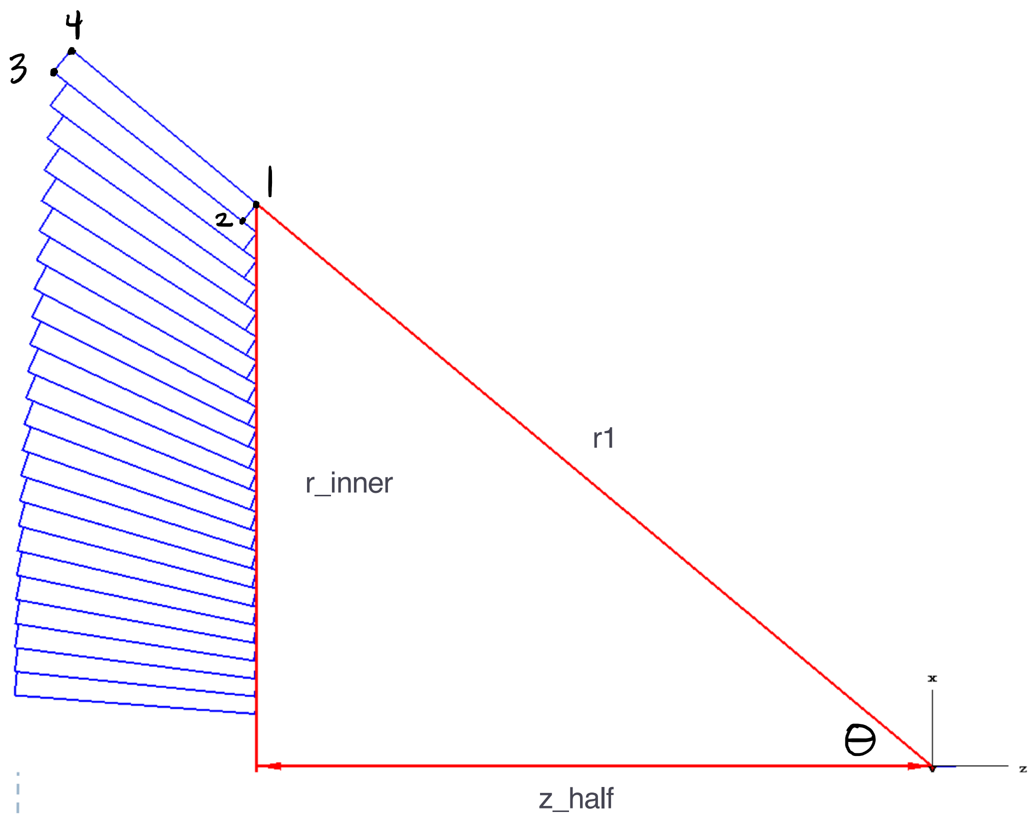
$$\boxed{\begin{aligned} z_4 &= z_1 \cdot \left(\frac{r_1 + t\_length}{r_1}\right) \\ x_4 &= x_1 \cdot \left(\frac{r_1 + t\_length}{r_1}\right) \end{aligned}}$$



Same logic for segmented ECAL towers (front and rear)

z5 = z2*(r2+t_length_front+t_length_rear)/r2
x5 = x2*(r2+t_length_front+t_length_rear)/r2

z6 = z1*(r1+t_length_front+t_length_rear)/r1
x6 = x1*(r1+t_length_front+t_length_rear)/r1



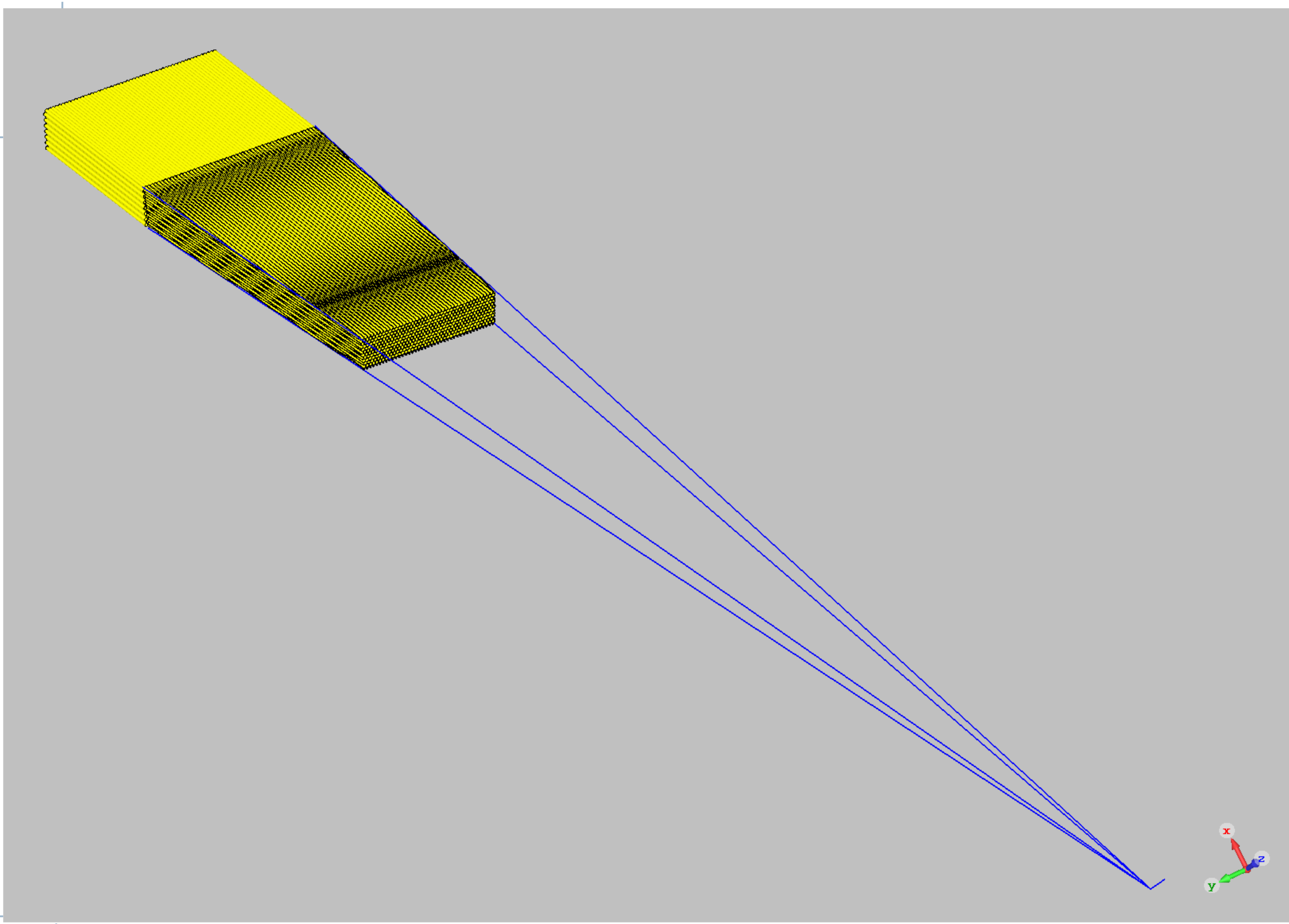
Endcap Towers

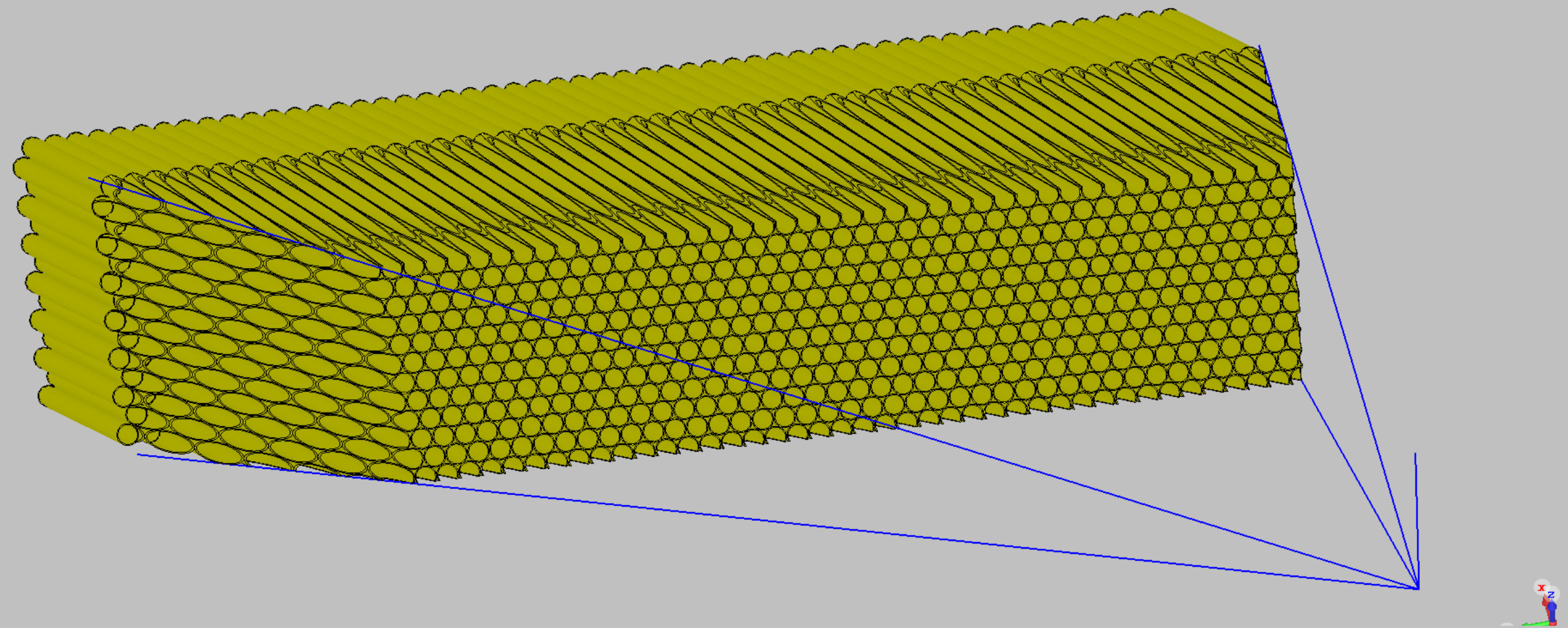
Same process follows for endcap towers

Z and X coordinates flipped
Theta extends from z axis, not x axis

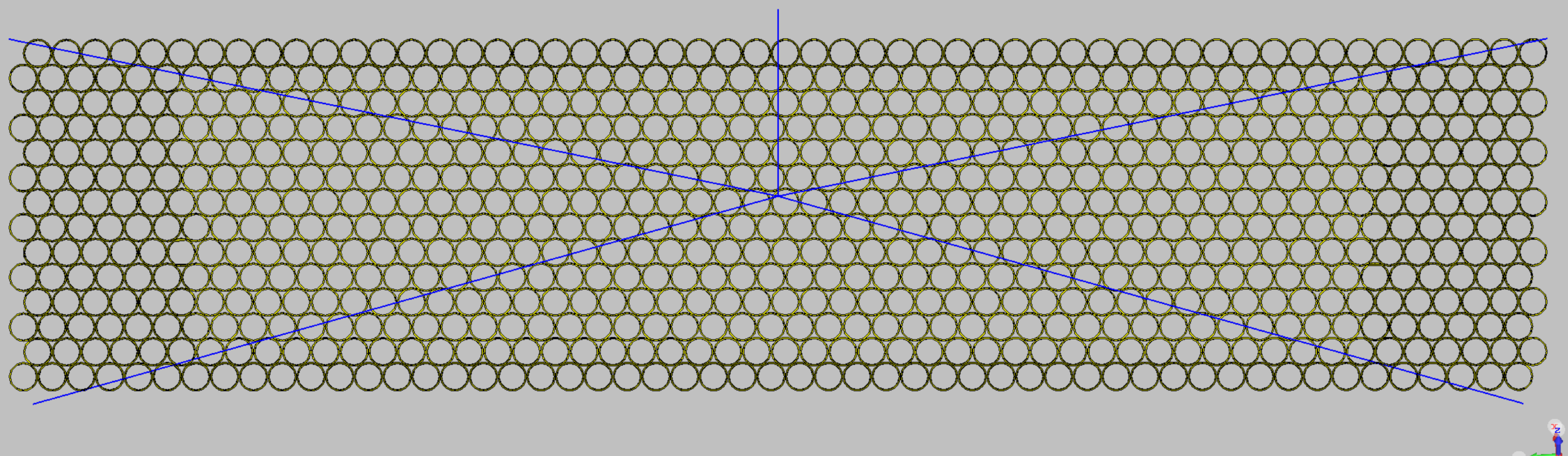
Direction of points 1, 2, 3, 4 is reversed

Leaves room for beampipe opening


x
z
y



x
z



x
z
y